Claim Amendments

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A method, comprising:

identifying instructions of a first thread that instruct a processor to perform a context swap to another thread; and

automatically inserting into the instructions of [[a]] the first thread at least one additional instruction that instructs the relinquishes control of a multi-tasking processor to perform a context swap to another thread based on counts of consecutively executed instructions of the first thread that do not instruct the processor to perform a context swap to another thread that will be concurrently sharing the processor .

2. (currently amended) The method of claim 1, further comprising:

identifying instructions of a second thread that instruct a processor to perform a context swap to another thread; and

automatically inserting into instructions of a second thread at least one instruction that instructs the relinquishes control of the multi-tasking processor to perform a context swap to another thread based on counts of consecutively executed instructions of the second thread that do not instruct the processor to perform a context swap to another thread that will be concurrently sharing the processor. ;

wherein processor is to swap execution among threads including the first thread and the second thread.

4

3. (currently amended) The method of claim 2, wherein

automatically inserting into instructions of the first thread comprises inserting

based on the identified instructions at least one characteristic of the instructions of the

second thread; and

automatically inserting into instructions of the second thread comprises inserting

based on the identified instructions at least one characteristic of the instructions of the

first thread.

4. (currently amended) The method of claim 2, further comprising:

repeating a procedure that determines one or more locations to automatically

insert instructions that relinquish control instruct the processor to perform a context

swap to another thread of the processor into the instructions of the first and second

threads.

5. (currently amended) The method of claim 3, wherein the automatically

inserting into instructions of the first thread based on the identified instructions at least

one characteristic of the instructions of the first thread comprises inserting into the

instructions of the first thread based on an average number of consecutive instructions

of the first second that do not relinquish control of the processor instruct the processor

to perform a context swap to another thread.

6. (currently amended) The method of claim [[5]] 3, wherein the automatically

inserting into instructions of the first thread based on the identified instructions ~~the at least one characteristic of the instructions of the first thread~~ comprises inserting into the instructions of the first thread based on a standard deviation derived from ~~the~~ a number of consecutive instructions of the second thread that do not ~~relinquish control of the processor~~ instruct the processor to perform a context swap to another thread.

7. (currently amended) The method of claim 1, further comprising:

constructing a data flow graph of the instructions of the first thread, the data flow graph comprising an organization of nodes associated with subsets of the instructions of the first thread; and

determining at least one of the following: a number of consecutive instructions ending [[a]] at one of the nodes that do not ~~relinquish control of the processor~~ instruct the processor to perform a context swap to another thread; a number of consecutive instructions beginning a one of the nodes that do not ~~relinquish control of the processor~~ instruct the processor to perform a context swap to another thread; and a number of consecutive instructions between instructions of one of the nodes that ~~relinquish control of the processor~~ instruct the processor to perform a context swap to another thread .

8. (original) The method of claim 1, wherein automatically inserting comprises inserting to keep intact a group of instructions identified as indivisible.

9. (original) The method of claim 1, wherein the processor comprises a multi-threaded central processor unit (CPU).

10. (original) The method of claim 1, wherein the processor comprises a multi-threaded engine of a multi-engine processor.

11. (currently amended) The method of claim 10, wherein the multi-threaded engine of the multi-engine processor comprises ~~an~~ a multi-threaded engine not having any floating point instructions in the multi-threaded engine's instruction set.

12. (currently amended) A ~~computer program product, disposed on a~~ computer readable medium ~~, the program including~~ storing instructions to cause a first processor to:

access instructions of a first thread;

identify instructions of the first thread that instruct a second processor to perform a context swap to another thread; and

insert into the instructions of [[a]] the first thread at least one additional instruction that instructs the second ~~relinquishes control of a multi-tasking~~ processor to perform a context swap to another thread based on counts of consecutively executed instructions of the first thread that do not instruct the second processor to perform a context swap to another thread ~~that will be concurrently sharing the processor~~.

13. (currently amended) The ~~program~~ computer readable medium of claim 12, further comprising instructions to:

access instructions of a second thread;

identify instructions of the second thread that instruct the second processor to perform a context swap to another thread; and

insert into instructions of a second thread at least one instruction that instructs the ~~relinquishes control of the~~ second processor to perform a context swap to another thread based on counts of consecutively executed instructions of the second thread that do not instruct the processor to perform a context swap to another thread ~~that will be concurrently sharing the processor.~~;

wherein second processor is to swap execution among threads including the first thread and the second thread.

14. (currently amended) The ~~program~~ computer readable medium of claim 13, wherein the instructions to:

insert into instructions of the first thread comprises inserting based on the identified instructions ~~at least one characteristic of the instructions~~ of the second thread; and

insert into instructions of the second thread comprises inserting based on the identified instructions ~~at least one characteristic of the instructions~~ of the first thread.

15. (currently amended) The ~~program~~ computer readable medium of claim 13, further comprising instructions to:

repeat a procedure that determines one or more locations to automatically insert instructions that ~~relinquish control~~ instruct the second processor to perform a context swap to another thread ~~of the processor~~ into the instructions of the first and second

threads.

16. (currently amended) The ~~program~~ computer readable medium of claim 14, wherein the instructions to cause to the first processor to insert into instructions of the first thread based on the identified instructions ~~at least one characteristic of the instructions of the first thread comprises an average number~~ instructions to cause the first processor to insert into the instructions of the first thread based on a count of consecutive instructions that do not ~~relinquish control of the processor~~ instruct the second processor to perform a context swap to another thread.

17. (currently amended) The ~~program~~ computer readable medium of claim 16, wherein the instructions to cause to the first processor to insert into instructions of the first thread based on the identified instructions ~~at least one characteristic of the instructions of the first thread comprises~~ comprise instructions to cause the first processor to insert into the instructions of the first thread based on a standard deviation derived from the number of consecutive instructions that do not ~~relinquish control of the processor~~ instruct the second processor to perform a context swap to another thread.

18. (currently amended) The ~~program~~ computer readable medium of claim [[1]] 12, further comprising instructions to cause the first processor to:

construct a data flow graph of the instructions of the first thread, the data flow graph comprising an organization of nodes associated with subsets of the instructions of the first thread; and

determine at least one of the following: a number of consecutive instructions ending [[a]] at one of the nodes that do not ~~relinquish control of the processor~~ instruct the second processor to perform a context swap to another thread; a number of consecutive instructions beginning a one of the nodes that do not ~~relinquish control of the processor~~ instruct the second processor to perform a context swap to another thread; and a number of consecutive instructions between instructions of one of the nodes that ~~relinquish control of the processor~~ instruct the second processor to perform a context swap to another thread.

19. (currently amended) The ~~program~~ computer readable medium of claim 12, wherein the instructions to cause the first processor to insert comprise instructions to insert to keep intact a group of instructions identified as indivisible.

20. (currently amended) The ~~program~~ computer readable medium of claim 12, wherein the processor comprises a multi-threaded central processor unit (CPU).

21. (currently amended) The ~~program~~ computer readable medium of claim 12, wherein the second processor comprises a multi-threaded engine of a multi-engine processor.

22. (currently amended) The ~~program~~ computer readable medium of claim 21, wherein the multi-threaded engine of the multi-engine processor comprises ~~an~~ a multi-threaded engine not having any floating point instructions in the multi-threaded engine's

instruction set.

23. (currently amended) The ~~program~~ <u>computer readable medium</u> of claim 22,

wherein the ~~program~~ <u>instructions comprise instructions</u> ~~comprises~~ <u>of</u> at least one of the

following: a compiler, an assembler, and a source code pre-processor.

24-29. (cancelled)